SDWb_Lua 密码登录案例使用说明

密码登录在设备界面授权功能上被广泛使用。本案例给出了如何使用 Lua 脚本来核验密码登录信息，整个过程无需用户单片机参与、无需串口数据交互，避免了通过串口发送密码登录信息，增加了密码登录信息管理的安全性。

一. 案例功能介绍

1.1 界面设计

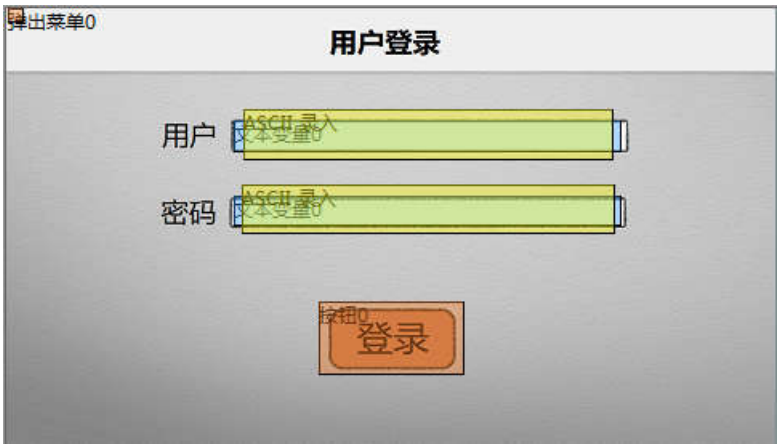

图 1 用户登录界面

典型的用户登录界面如图 1 所示，用到了 ASCII 录入、文本显示、按钮和弹出菜单四种控件。在界面中，用户首先需要输入用户名及密码，然后点击“登录”按钮。

点击“登录”按钮，有两种方法可以实现对输入的用户和密码进行校验。一种方法是用户单片机通过串口读取到输入的用户和密码信息，并将其与预置存储的信息进行比对。另一种方式是通过 Lua 脚本编程，直接将输入的用户和密码信息，与 Lua 脚本中预置的信息进行比对。后一种方法无需通过串口进行信息交互，登录密码信息管理更加安全。下文将详细介绍如何用 Lua 脚本编程核验输入的用户和密码信息。

1.2 控件设计

在图 1 中，用户名和密码信息的录入，需要使用文本变量控件和 ASCII 录入控件组合实现。使用 ASCII 录入控件录入数据，再通过文本变量控件将录入的数据显示出来，此过程不需要 lua 脚本参与。

“登录”按钮用于触发登录操作，此按钮需要设置按键键码，以便在触摸事件发生时，能够执行 Lua 中的触摸回调函数，通过页面编号以及按键键码，可以在 Lua 代码中定位到该按钮控件。登录按钮的按键键码设置如图 2 所示。

弹出菜单控件用于提示用户名或密码录入错误。此控件同样需要设置按键键码，但

是功能与登录按钮完全不同，该按键键码用于软件方式触发弹出菜单功能。弹出菜单的按键键码设置如图 3 所示。

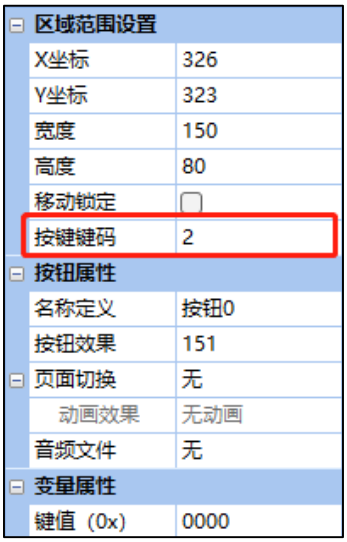

图 2　登录按键的键码值

图 3　弹出菜单的键码值

1.3 流程设计

登录流程如图 4 所示。因为“登录”按钮设置了按键键码，所以当点击“登录”按钮时会触发 Lua 触摸回调函数的执行。在 Lua 代码中，首先从变量存储器中读取用户名和密码数据，然后将读取的数据与 Lua 中预置的用户名和密码进行比较。如果完全相等，则跳转到主页面表示登录成功并清除输入的密码；如果不相等，则触发弹出菜单，提示用户名或密码错误。

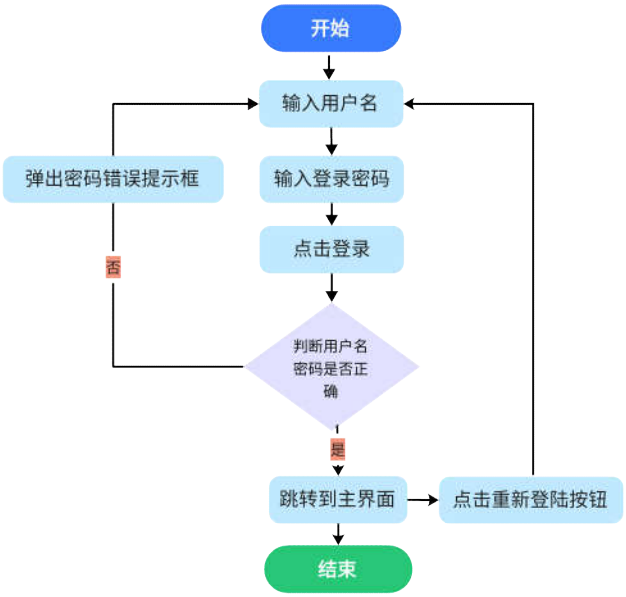

图 4　密码登录流程图

二．初始化回调函数

该案例中，可以在初始化回调函数中设置串口相关参数。

```
--初始化回调函数
function callback_init()

    --将串口0设置为VGUS协议，波特率115200，串口格式8N1。
    com_set_work_mode(0,0,115200,4)

    --将串口1设置为自定义串口协议，波特率115200，串口格式8N1。
    com_set_work_mode(1,1,115200,4)

    --选择调试串口为串口0。（默认为串口1）
    com_set_debug_print(0)

    --打印调试信息
    print("callback_init;\r\n")

end
```

该案例也可以使用默认设置，不调用初始化回调函数。

三．触摸回调函数

点击“登录”按钮后，如图 3 所示，因为该按钮设置了“按键键码”为 2，所以触发按键后串口屏开始调用触摸回调函数 callback_touch，在回调函数下用 if 语句来判断是否触发该按键，当满足条件时开始执行嵌套在 if 语句里的代码。

```
--150 号页面，按键键码为 2，按键状态为松开状态

if pic_id == 150 and key_code == 2 and touch_state == 2 then
```

当上述条件满足时，也就是 150 号页面的登录按钮被触发，且按钮松开了，开始执行以下代码。

3.1 读取用户名、密码数据

这里用户名以及密码所设置的文本变量控件设置的变量地址分别为 0xD100 和 0xD200,所以读取用户名以及密码的代码如下：

```
--以字符串方式读取变量存储器中的数据

read_name=vgus_vp_string_read(0xD100)

read_password=vgus_vp_string_read(0xD200)
```

这里使用以字符串方式读取变量函数 vgus_vp_string_read(vp_addr)分别读取用户和密码的文本数据，并赋值给两个字符串类型的变量“read_name”以及“read_password”。

3.2 判断密码是否正确

本程序在开头定义了用户名以及密码，其代码如下：

```
local user_name = "admin"
```

```
local user_password = "abcdef"
```

判断密码是否正确，就是判断“read_name”以及“read_password”是否分别等于“user_name”以及“user_password”,这里用 if 语句就可以判断，其代码如下：

```
if read_name == user_name and read_password == user_password then
```

3.3 密码正确

若用户和密码正确，切换到 0 号页面，其代码如下：

```
--切换到 0 号页面
reg_data[1]=0x00;
reg_data[2]=0x00;
vgus_reg_write(0x03, 2, reg_data)
```

因为 03，04 寄存器控制当前显示的页面 ID，所以通过写寄存器函数 vgus_reg_write(reg_addr, write_len, write_table),从 03 寄存器开始，往 0x03 以及 0x04 寄存器里写入 0x00、0x00 可以跳转到 0 号页面。

清除密码数据，其代码如下：

```
--清除密码数据
for i=1,10,1 do
    vp_data[i]=0xFFFF;
end
vgus_vp_write(0xD200, 10, vp_data)
```

首先给定义的一个数据表 vp_data 赋值，需要注意的是 API 接口函数的 table 参数都是从 1 开始索引的，通过 for 循环将 vp_data[1]~vp_data[10]都赋值为 0xFFFF，然后通过写变量存储器函数 vgus_vp_write(vp_addr, write_len, write_table) 将 0xD200-0xD209 全部赋值为 0xFFFF，达到清空密码的目的。

3.4 密码错误

若用户和密码错误，则执行弹出菜单提示密码错误，其代码如下：

```
--使用弹出窗口控件，弹出提示错误提示信息
reg_data[1]=0x01;
vgus_reg_write(0x4f, 1, reg_data)
```

弹出菜单控件设置的按键键码为 1，这里给 reg_data 的第一个数 reg_data[1]赋值 1，

然后通过写寄存器函数 vgus_reg_write(reg_addr, write_len, write_table)往用于触发触控配置的寄存器 0x4F 寄存器里写入 reg_data[1],也就是按键键码为 1，这样就可以触发弹出菜单，提示密码错误。

以上是实现密码登录的基本步骤，官网可以下载完整的案例工程，包括 Lua 脚本代码和界面工程。控件的使用说明、寄存器功能说明都可以参考文档《VGUS 串口屏用户开发指南》。Lua 脚本函数可以参考文档《基于 VGUS 的 Lua 脚本使用说明》。